

**INSTITUTO TECNOLÓGICO DE COSTA RICA
VICERRECTORÍA DE INVESTIGACIÓN Y EXTENSIÓN
DIRECCIÓN DE PROYECTOS**

INFORME FINAL DE PROYECTO DE INVESTIGACION

**Desarrollo de herramientas para la visualización de modelos
de datos científicos**

Código 5402-1375-1701

ELABORADO POR

Armando Arce Orozco, M.Sc.

Escuela de Computación

Noviembre, 2015

Tabla de Contenidos

I. DATOS GENERALES	3
I. INTRODUCCION	4
II. MARCO TEORICO	6
III. METODOLOGIA	13
IV. RESULTADOS	15
V. DISCUSION Y CONCLUSIONES	17
VI. RECOMENDACIONES	19
VI. REFERENCIAS	21

I. DATOS GENERALES

1. Nombre del Proyecto

Desarrollo de herramientas para la visualización de modelos de datos científicos

Código: 5402-1375-1701

2. Autores

Coordinador: Armando Arce Orozco M.Sc. arce@itcr.ac.cr

3. Resumen:

La visualización científica es clave para desarrollar el aprendizaje. Los modelos digitales sirven para expresar, probar y optimizar diseños, pero para usarlos efectivamente se requiere vincular lo que se aprende en el mundo artificial con lo que se aprende en la realidad. Los nuevos esquemas de diseminación de datos en formatos vectoriales, como la transmisión incremental, plantean una solución factible al problema de manipular y visualizar estructuras digitales complejas a través de Internet. El uso de "mosaicos" de imágenes estáticas generadas a partir de modelos 3D, podría ser también una opción útil para la visualización distribuida de cierto tipo de modelos altamente complejos. Hasta el momento, la gran mayoría de productos de visualización de datos se han concentrado en los aspectos geométricos del modelo por lo que la mayoría de los formatos de transferencia vectorial no cuentan con mecanismos que permitan la vinculación de atributos a dicho modelo geométrico. Esto representa una gran desventaja pues al examinar un modelo 3D es necesario, para diferentes tareas de análisis, conocer cuáles son los datos cualitativos o cuantitativos que están asociados a una zona particular del modelo. Dichas capacidades generalmente están presentes en paquetes de visualización científica específicos, pero no son soportadas cuando se necesita visualizar modelos de forma remota. El enfoque de este proyecto consiste en la integración de tres herramientas computacionales específicas, para facilitar la diseminación de modelos de fenómenos científicos entre el público en general.

4. Palabras claves

Visualización, visualización científica, modelos de datos digitales, OpenDX, JavaView

I. INTRODUCCION

La visualización científica es una clave para desarrollar el aprendizaje. Los modelos digitales sirven para expresar, probar y optimizar diseños; pero para usarlos efectivamente se requiere vincular lo que aprende de forma virtual con lo que se aprende en la realidad. Cuando se observan modelos digitales, se debe ser capaz de ver no sólo lo que está sobre la pantalla, sino de imaginar el fenómeno físico que tal modelo representa.

Se dispone de cantidades masivas de datos de fenómenos científicos de todas las clases que podrían ser de fácil acceso para los estudiantes, por medio del uso de herramientas sencillas de visualización. Es necesario desarrollar este tipo de herramientas, para servir al sistema educativo, con el fin de ayudar a mejorarlo y hacerlo eficiente. Además, estas herramientas ayudarían también a resolver el problema de cómo lograr que los estudiantes digieran y comprendan todo el cúmulo de conocimientos que estudian. La clave para esto es la ayuda de la tecnología avanzada aplicada a la educación.

El propósito de la visualización científica es el entendimiento profundo de fenómenos científicos. Hasta ahora el tipo de audiencia, para esta tecnología, ha sido muy especializada y altamente técnica. Sin embargo, esto puede cambiar si se cuenta con herramientas que faciliten la diseminación y visualización masiva de estos modelos de datos científicos.

El enfoque de este proyecto consiste en la integración de tres herramientas computacionales específicas, para facilitar la diseminación de modelos de fenómenos científicos entre el público en general. Estas tres herramientas incluyen: un sistema de visualización científica poderoso (Data Explorer), un visualizador de modelos geométricos pequeño y eficiente (JavaView), y un formato estandarizado para realizar la transferencia de los modelos científicos (XML).

Data Explorer es un sistema de herramientas e interfaces de usuario para visualizar datos científicos. La interfaz gráfica de Data Explorer permite al usuario ejecutar tareas de varios niveles de sofisticación. En Data Explorer una imagen es creada aplicando datos, como entradas, a un programa visual. El programa visual es una secuencia de funciones interconectadas (módulos) que actúan sobre una o mas

entradas, y producen una o mas salidas. Típicamente, la entrada son datos del usuario y la salida es una realización de los datos en forma de una imagen.

Data Explorer provee un extenso y poderoso conjunto de módulos de visualización altamente interoperables. Por ejemplo, los módulos Isosurface, Streamline y AutoColor ejecutan las funciones de visualización estándar como crear superficies de valores constantes, seguimiento de la trayectoria de partículas a través de un campo vectorial, y coloración de objetos basado en el valor de sus datos, respectivamente.

El modelo de datos de Data Explorer se centra en la noción de campos continuos (Fields) . En Data Explorer, un campo continuo representa un mapeo desde algún dominio a algún espacio de datos. El dominio del mapeo es especificado por un conjunto de posiciones y (generalmente) un conjunto de conexiones que permiten la interpolación de los valores de los datos, para puntos en el dominio entre posiciones específicas. El mapeo de todos los puntos en el dominio está representado implícitamente por datos específicos que son dependientes de (localizados en) los puntos o en las conexiones entre los puntos.

Por su parte, JavaView es un visualizador de geometrías en 3D y un programa de visualización matemática. Su integración con el Web permite desplegar geometrías en 3D y experimentos geométricos interactivos en cualquier documento HTML en el Internet. JavaView también puede ejecutarse como una aplicación en computadoras basadas en Unix o MS Windows. Una versión más pequeña de JavaView, llamada JavaView-Lite, ha sido desarrollada y optimizada para ser descargada rápidamente mediante el Web. Esta versión es utilizada para desplegar modelos de geometrías precalculadas, dentro de páginas Web, tales como objetos sólidos.

II. MARCO TEORICO

La visualización es el proceso de representar datos mediante una imagen visual. Sin embargo, el propósito de la visualización es la comprensión, no las imágenes por sí mismas.

Particularmente, las tecnologías de visualización tratan con el uso de representaciones visuales de datos, en forma interactiva y soportadas por computadora, para aumentar la cognición. En estas, la presentación de datos usa representaciones espaciales o gráficas, que facilitan la comparación, reconocimiento de patrones, detección de cambios y otras habilidades cognitivas que hacen uso del sistema visual.

1. Definiciones

La visualización científica trata con la transformación de datos físicos, producidos a través de cálculos o experimentos científicos o de ingeniería, en imágenes. [Schoroeder, Martin, y Lorensen, 1998].

Este tipo de herramientas representan un proceso gráfico análogo al análisis numérico, y es a veces referido como análisis visual de datos. La visualización científica permite que cualquier objeto físico sea visualizado.

El propósito de la visualización científica es el entendimiento profundo de fenómenos científicos. El tipo de audiencia es especializada y altamente técnica. Entre las fuentes de datos físicos en que tiende a basarse se encuentran el cuerpo humano, la atmósfera, la tierra, moléculas y otras.

2. Estructura del sistema de visualización

Muchos modelos de referencia, para sistemas de visualización científica, han sido propuestos en la literatura. Estos varían en su alcance, objetivo y audiencia. Aquí se presentará el modelo propuesto por Haber y McNabb [Haber y McNabb, 1990] que provee una útil clasificación general desde el punto de vista del usuario. El modelo de Haber y McNabb divide la visualización en tres procesos generales, cada uno actúa en algunos datos para producir un nuevo conjunto de datos.

El primer proceso, la creación de un modelo de datos, actúa sobre datos crudos. Este proceso crea un modelo de los datos en que un nuevo conjunto de datos derivados puede ser producido. Este opera en datos crudos y los modifica en una o mas formas para derivar datos para operaciones subsecuentes. Este abarca procesos tales como la interpolación y el cálculo de cantidades derivadas. Partes de este proceso no son interactivas.

El segundo proceso, llamado realización, crea uno o mas objetos imaginarios llamados Objeto de Visualización Abstracto (OVA) desde los datos derivados producidos por el proceso previo. Cada cantidad en el dato derivado es asociado a un atributo del OVA. Esta es una parte central del proceso de visualización científica, y generalmente interactivo.

El tercer proceso, la generación (rendering), es donde una o mas vistas del OVA son generadas para producir una imagen en un dispositivo de salida, generalmente la pantalla de la computadora. Este comúnmente utiliza las capacidades de gráficas de la computadora, transformaciones geométricas, remoción de superficies ocultas, iluminación y sombreado, aunque algunas técnicas han sido desarrolladas especialmente, tal como la generación de volúmenes.

3. Dominio Espacial

Esta sección presentará algunas de las técnicas que se pueden utilizar para representar el dominio espacial en los campos continuos, de las cuales Goodchild [Goodchild, 1992] identifica seis tipos diferentes: rejillas de celdas, teselado poliédrico, complejos simpliciales, rejillas de puntos, y puntos irregulares.

Rejillas de celdas

Una rejilla de celdas es un particionamiento del área de estudio V en celdas $S[i]$. Los planos frontera de las celdas son perpendiculares o paralelos unos con otros. Una celda puede definirse como el cerco convexo [Preparata y Shamos, 1985] de $2n$ vértices de un lattice. En el caso de $\dim D = 1$, las celdas realmente son líneas, para $\dim D = 2$ las celdas son cuadriláteros (quads), y para $\dim D = 3$ las celdas representan paralelepípedos.

Teselado poliedrico

Un teselado poliedrico es un teselado del área de estudio en poliedros $S[i]$ de dimensión $\dim D$, o sea, los poliedros tienen igual dimensión que D . Esta es una generalización de la rejilla de celdas mencionadas anteriormente en el sentido de que los $s[i]$ son generalmente dados como el cerco convexo de un conjunto de m puntos, donde $m > \dim D$. Sin embargo, los poliedros no necesitan ser convexos. Si $\dim D = 2$, esta es llamada un cobertura poliedrica. En general se asume que dos poliedros $s[i]$ y $s[j]$ de un teselado se encuentran, si no del todo, en una cara o arista común, respectivamente.

Complejos simpliciales

Este es un caso especial de un particionamiento poliedrico con $m = \dim D + 1$. Cada $s[i]$ es un m -simplex, o sea, el cerco convexo de m puntos. En el caso de $\dim D = 1$, los simplicies son líneas (al igual que en la rejilla de celdas), para $\dim D = 2$ los simplicies son triángulos y el complejo simplicial es a veces llamado red irregular triangulada (TIN, por sus siglas en inglés). Cuando $\dim D = 3$, los simplicies representan tetraedros. El conjunto de simplicies $s[i]$ algunas veces obedece a condiciones adicionales tal como el criterio de Delaunay [O'Rourke, 1994]. Nótese que todo el simplex está relacionado al valor del campo y no solo los vértices del simplex como es lo usual en una TIN.

Rejilla de puntos

Los índices $s[i]$ son aquí el subconjunto de un lattice con vectores generadores $v[1], \dots, v[m]$. Los índices por tanto contienen exactamente un punto de V . Sin embargo, la estructura es equivalente (homeomórficamente) a la rejilla de celdas ya que cada punto del lattice puede ser mapeado a un celda correspondiente dada por el cerco de $2m$ puntos.

Puntos irregulares

Esta es una generalización de un lattice. Los $s[i]$ son puntos de V , sin un orden implícito. Una equivalencia puede ser definida con un teselado poliedrico y con un complejo simplicial. El mapeo de un teselado poliedrico es dado por el correspondiente diagrama de Voronoi [Preparata y Shamos, 1985]. Un mapeo biyectivo a un complejo simplicial no es siempre posible. Si existen algunas irregularidades en el conjunto de $s[i]$ entonces no existe una triangulación de Delaunay no ambigua.

4. Realización

La realización consiste en crear una representación geométrica, esto es, convertir el modelo de datos en algo que pueda ser dibujado. Esta sección discute algunas técnicas de visualización comunes.

Contornos e isosuperficies.

Dado un conjunto de datos recolectados sobre una región continua, es común considerar el dibujar líneas suavizadas que conecten juntas la posiciones que contienen datos con los mismos valores, a este tipo de líneas se le llama líneas de contorno o isolíneas. La técnica de contornos presenta una serie de isolíneas que incluyen todos los puntos de un conjunto de datos. En una gráfica de contorno en dos dimensiones, el eje Z es perpendicular al plano y por tanto todas las líneas de contorno aparecen en el mismo plano.

Cuando existen muchas líneas de contorno, cada quinta línea u otra, son resaltadas para ayudar al observador. Si lo que se desea es poder representar el valor de una tercera variable Z en el plano, muchas veces se rellena el área entre líneas de contorno con un color o patrón particular asociado a los valores de dicha variable. A esto se le llama una gráfica de contorno rellena.

La técnica de contornos puede ser generalizada a un dominio tridimensional resultando en la técnica de isosuperficies. Una isosuperficie, en tres dimensiones, conecta todos los puntos que tienen el mismo valor en un conjunto de datos determinado. En estas gráficas las áreas entre las líneas que conectan los puntos de igual valor, son presentadas opacas. Al igual que en el caso de las gráficas de contorno, se puede agregar color o brillo, resultando en las llamadas gráficas de isosuperficies rellenas.

Campos vectoriales

Los conjuntos de datos con valores vectoriales ocurren muy frecuentemente en visualización científica. La técnica mas sencilla para representar campos vectoriales con los "glyphs" vectoriales. Un "glyph" es una figura o imagen generada a partir de los valores de variables particulares. Cada ocurrencia del "glyph" representa un valor único de la variable asociada. Algunos atributos del "glyph" (p. ej. largo o ángulo) son una función de la variable y varían con él. Los "glyphs" vectoriales comúnmente

lucen como flechas o cohetes. El campo vectorial es muestreado en un conjunto específico finito de localizaciones. En cada localización, una flecha es construida con su largo y dirección determinadas por los componentes del vector. Esta no es realmente una representación continua, debido a que solo un conjunto finito de puntos es muestreado. Si se incluyen suficientes puntos, relativos a la geometría del dominio y las gradientes del campo vectorial, entonces el usuario pueden mentalmente interpolar entre los puntos muestreados para llenar el campo continuo.

Los “glyphs” vectoriales proveen una representación pictórica directa del campo vectorial mismo y tienden a ser independientes de la aplicación. Sin embargo, un problema asociado a esta técnica es que si hay muchos puntos o la longitud de la flecha es muy larga, los “glyphs” pueden ocultarse entre sí.

Volúmenes

La generación (rendering) de volúmenes es una técnica utilizada para examinar datos recolectados en medio del espacio definido por un volumen. Esta técnica utiliza color y transparencia para visualizar todos los datos en el conjunto de datos. Los detalles internos visualizados puede ser físicos (tal como la estructura de las partes de una máquina, o huesos y músculos en el cuerpo humano) o ellos puede ser otras características (tales como flujo de fluidos, o temperatura).

Originalmente el volumen es transparente. Conforme la luz pasa a través del mismo, este absorbe mas en áreas en donde se concentran los valores. Estas áreas parecen ser mas opacas. Si se asigna color a cada valor se verán áreas coloreadas de distintos colores en relación con otras.

Otra técnica para visualizar volúmenes, que no requiere el uso de la transparencia, es el plano de corte [DeFanti, Brown, y McCormick, 1989]. Un plano de corte divide un volumen en una región visible y otra no visible. Esta técnica hace posible ver datos escalares en una sección de corte del volumen. En este caso, inicialmente se define un rejilla regular cartesiana en el plano de corte y luego los valores de los datos en esta rejilla son encontrados por interpolación de los datos originales en el volumen. El uso conveniente del color hace visible los datos del volumen reflejados al plano, haciendo el interior del objeto visible.

Trayectorias de partículas

El movimiento de partículas en un campo vectorial puede ser representado mediante pequeños puntos en movimiento, esta técnica es llamada animación. La animación simula movimiento continuo desplegando rápidamente imágenes. El usuario tiene la impresión que él está mirando un movimiento continuo. Para lograr esto, el hardware gráfico necesita desplegar imágenes a una tasa superior a 25 imágenes por segundo, sino el movimiento puede lucir abrupto.

Otras técnicas utilizan líneas derivadas de las trayectorias de las partículas. La técnica de rayas (streaklines), también llamadas rastros, utiliza líneas para representar el camino de las partículas a través de un campo vectorial cambiante. La técnica de líneas de corriente (streamlines), también llamadas líneas de flujo, utiliza líneas para representar el camino de las partículas en un campo vectorial en un momento particular.

Una vez que se ha generado una línea derivada, se pueden utilizar técnicas adicionales para resaltar otras características de las trayectorias. Una de estas técnicas son las cintas (ribbon), que consisten de un plano angosto que sigue el recorrido de la línea derivada. Dicho plano puede rotar para mostrar características como la vorticidad. Otra técnica consiste en generar tubos (tubes), que son volúmenes que siguen el recorrido de la línea derivada. El tubo también puede rotar para mostrar características adicionales.

5. Generación (Rendering)

La generación de imágenes involucra el establecimiento de parámetros de luminosidad en cada superficie visible de los objetos en una escena en tres dimensiones, como se vería desde el punto de visión del usuario. En forma similar, en imágenes en dos dimensiones, la generación es importante para seleccionar el grosor de las líneas, los factores de acercamiento, etc.

El proceso de generación de una imagen involucra el cálculo por la computadora de la cantidad de luz que recibe cada superficie visible de los objetos en la escena, a como es visto desde el punto de visión de la "cámara" de la computadora (el punto de vista del usuario). Durante el proceso de generación, las propiedades de las superficies de los objetos son tomadas en cuenta, tales como son los colores y las luces brillando sobre los objetos. En otras palabras, un gráfico de computadora

realiza un escrutinio de la escena frente a la cámara, a la resolución del monitor de la computadora, en que la escena será desplegada.

Realmente la generación de la imagen no crea una imagen en tres dimensiones; esta sólo calcula los colores de los puntos que pueden ser vistos en el pantalla, en dos dimensiones del monitor, desde el punto de visión elegido. Las partes de los objetos que no pueden ser vistas, no son escrutados, ni generadas, ni almacenadas en la imagen, ni desplegadas en el monitor. Esta imagen en dos dimensiones puede parecer de tres dimensiones a los ojos del observador debido al sombreado, la oclusión de objetos distantes por otros mas cercanos, y otras técnicas visuales que, en el mundo real, indican dimensionalidad.

III. METODOLOGIA

Con el fin de realizar la programación, depuración y pruebas de los diferentes productos generados en este proyecto se dividieron las actividades en las siguientes:

1. Revisión de antecedentes.

La etapa de revisión de antecedentes consistió de la obtención de todo aquel material relacionado al tema de estudio y el análisis de los diferentes enfoques utilizados para resolver el problema en cuestión. En esta actividad se hizo una revisión de cómo el Web pueden ser, y están siendo, utilizado para resolver el problema de acceder, almacenar y procesar información científica.

2. Integración de aspectos teóricos

En la etapa de integración de aspectos teóricos se llevó a cabo la adaptación de los diferentes conceptos relacionados con la teoría en la que se fundamenta el diseño de sistemas de visualización científica. Una vez clasificada y estructurada esta información se procedió a la redacción de la sección del documento que resumieran las estructuras y modelos de datos científicos, así como los estándares para transferencia y documentación.

3. Estudio de software y diseño de programas

Esta etapa requirió del estudio de la herramienta de visualización seleccionada (OpenDX), la cual fue adaptada para incorporar los modelos científicos. En esta misma etapa se incluyó el estudio de la forma de operación del mismo, junto con los mecanismos para extender la funcionalidad de dicho sistema. Posteriormente, se hizo una revisión a fondo del modelo de datos científico utilizando por OpenDX. Como resultado se redactó la partes del documento que resume todos los conceptos principales de este modelo.

4. Diseño, implantación y depuración de programas

En la etapa de diseño de módulos se realizó el diseño de los módulos de software necesarios. Aquí se decidió adaptar un software existente "httpd" en lugar de

desarrollar un servidor web desde cero. También se probó y adaptaron los módulos que se comunican con el API de OpenDX. La posterior etapa de implantación requirió la codificación, utilizando el lenguaje (o mecanismo) provisto por el sistema de visualización científica y lenguaje C, de cada una de las entidades y procesamiento asociado, definido en la etapa de diseño. Luego de la implantación, se procedió con la etapa de depuración y pruebas de las rutinas codificadas.

5. Análisis de resultados y etapa de conclusiones.

En la etapa de análisis de resultados se realizó una valoración objetiva de la funcionalidad provista por la herramienta. En particular, se evaluó en forma empírica el rendimiento de la herramienta para modelos de datos grandes y complejos. Conforme se encontraron debilidades se necesitó realizar modificaciones a los programas desarrollados. Para finalizar, la etapa de elaboración de conclusiones permitió realizar una reflexión sobre los objetivos alcanzados, y sobre los posibles desarrollos futuros.

IV. RESULTADOS

Actualmente no existen estándares ampliamente utilizados para la disseminación de datos científicos, así como existen muy pocas herramientas de uso público que permitan la exploración y navegación de grandes cantidades de estos tipos de datos. Además, la transferencia y visualización distribuida de modelos de datos científicos representa un gran reto pues la cantidad de información es muy grande como para transferirla rápidamente en formato vectorial, a través de una red como Internet; y por otra parte la interacción dinámica y rápida que requieren los usuarios impide utilizar un esquema basado en transferencia de imágenes estáticas.

En este proyecto se evaluaron dos diferentes soluciones: una que utilizaba las capacidades existentes del producto *OpenDX* para generar imágenes a partir del modelo 3D; y otra solución que exportaba el modelo a un formato vectorial que luego sería visualizado mediante el “plugin” especial (*JavaView*).

En ambos casos anteriores, los productos desarrollados permitieron la visualización distribuida de los modelos a través de Internet y facilitaron la interacción del usuario con dichos datos. Sin embargo, encontramos una serie de inconvenientes en ambas implementaciones:

Visualización mediante imágenes

Por otra parte, el hecho que cada solicitud consiste en generar una nueva imagen por parte del servidor, también representa una carga excesiva del mismo. Para evitar este problema se modificó la configuración del servidor para incluir un mecanismo de “cache” de solicitudes de la misma forma que lo realizan los servidores httpd. Sin embargo, este esquema no fue totalmente exitoso debido a la cantidad tan grande de valores diferentes que pueden tomar los parámetros al realizar la solicitud de visualización..

La visualización de modelos 3D mediante imágenes presentó un bajo rendimiento para modelos complejos. En particular, este fenómeno sucedió debido a que el software servidor (*OpenDX*) requiere la carga de todos los archivos necesarios, cada vez que se realiza una solicitud por parte de los clientes. Una modificación realizada

al proceso de generación permitía que los modelos se encontraran “precargados” a la hora de solicitar las diferentes imágenes, lo que mejoró bastante el tiempo de ejecución; sin embargo, esta no parece una solución escalable pues cada modelo requiere una cantidad considerable de memoria y es poco factible tener un gran número de modelos complejos cargados al mismo tiempo.

Adicionalmente, el mecanismo de consulta de los atributos asociados al modelo requirió del desarrollo de algunos procedimientos poco eficientes, que utilizan la escritura en disco por parte del servidor, para poder enviar la información a los procesos clientes.

Visualización mediante vectores

La visualización de modelos mediante vectores presentó también una serie de inconvenientes. En general, el principal de ellos fue la imposibilidad de cargar rápidamente modelos grandes y complejos. De hecho, para modelos de un tamaño superior al megabyte, el proceso de visualización fue demasiado lento, o bien, la carga inicial era lenta y la posterior visualización si era más rápida. Se utilizaron diferentes formatos para probar (en forma empírica) este esquema y se obtuvieron rendimientos similares para formatos binarios y para formatos de texto (xml) comprimidos (zip).

La herramienta utilizada para la visualización en el cliente (*JavaView*) requiere la instalación del ambiente de ejecución Java en la máquina del usuario. Esto no siempre es factible debido a que no todas las personas conocen o están dispuestas a realizar la instalación de un software de este tipo, sin contar con el hecho que el archivo de instalación es bastante grande (aproximadamente 7 megabytes).

Además, otro problema que presentaba la herramienta utilizada (*JavaView*) es que por no ser “open source” era prácticamente imposible modificarla para que se pudiera transferir y consultar los atributos asociados al modelo. Originalmente, se pensó que bastaría con programar un par de rutinas utilizando el API provisto por la herramienta, sin embargo, conforme se avanzó en el proceso de desarrollo se determinó que dicho API era insuficiente para desarrollar las capacidades necesarias.

V. DISCUSION Y CONCLUSIONES

El haber desarrollado y codificado una serie de rutinas de software para visualización de datos científicos, no es el único aporte relevante de esta investigación. Aún cuando dichas rutinas representan el producto final de este proceso y corresponden a la aplicación práctica de los principios teóricos presentados al inicio de este documento, durante la realización de este trabajo se obtuvieron una serie de aportes adicionales que son importantes de resaltar.

1. Generales

Este trabajo incluye una breve revisión de antecedentes históricos que pueden representar un punto de inicio en cualquier investigación sobre esta área en particular. Al analizar dichos antecedentes es posible notar el hecho que las tecnologías de visualización científica están fuertemente relacionadas con otro tipo de disciplinas como son la estadística, el diseño gráfico, los sistemas de bases de datos, y otras. Esta relación es importante porque cada disciplina se puede ver beneficiada de los adelantos que se realicen en las otras áreas.

Otro aporte importante, que presenta este trabajo, es haber realizado la identificación de los requerimientos que un sistema de visualización científica debe satisfacer. Estos requerimientos, aún cuando solo abarcan un conjunto mínimo de funcionalidades, son importantes para comprender y definir el grado de alcance de este tipo de sistemas.

Adicionalmente, este proyecto ha presentado un diseño totalmente independiente de la codificación, que puede ser utilizado para desarrollar un sistema básico para visualización de datos científicos. Este diseño incorpora todos aquellos mecanismos que son de esperar en cualquier herramienta de este tipo. Aquellas personas o empresas, que deseen iniciar el desarrollo de una de estas herramientas, se pueden beneficiar grandemente al analizar el diseño planteado y proponer modificaciones o ampliaciones a este.

2. Académicos

Un aspecto que puede ser de gran interés para la comunidad académica, y especialmente en lo que se refiere al desarrollo de investigaciones, es la revisión que se ha realizado sobre los mecanismos que utiliza la visualización científica para representar datos espaciales. Científicos en diferentes áreas como mecánica de fluidos, meteorología, geología, y muchas otras; se pueden beneficiar al comprender los principios en que se basan este tipo de herramientas. Comprender estos principios es el primer paso antes de iniciar el uso eficiente de los sistemas de visualización científica en cualquier disciplina.

Sin embargo, más valiosa aún, es la síntesis de los fundamentos teóricos de la visualización científica que se ha presentando en este trabajo. En gran parte estos fundamentos son generales para cualquier tipo de representación visual. Por tanto, no solo los profesores y estudiantes del área de la computación pueden sacar provecho de estos principios, al diseñar o desarrollar interfaces para aplicaciones que requieren el acceso a grandes modelos de datos digitales, si también en otras disciplinas como la física, la biología y la medicina.

Por otra parte, el mayor aporte práctico de este proyecto es que se ha desarrollado un conjunto de herramientas que permiten generar visualizaciones que son visibles de forma remota. Dicha herramienta puede ser de gran ayuda al desarrollar cursos universitarios en los que se requiere visualizar grandes modelos digitales complejos de forma eficiente y efectiva. Esta herramienta brinda la posibilidad de que el estudiante aplique directamente los principios teóricos aprendidos en clase, al realizar visualizaciones de forma sencilla pero flexible.

Por último, debido a que estas herramientas se basan en un paquete de software de uso gratuito, la posibilidad de que más personas puedan utilizarlo se incrementan. La herramienta está disponible en muchas plataformas Unix y recientemente se han desarrollado versiones para plataformas Windows. Esto hace que la base de usuarios potenciales se amplíe aún más. Como ya se señaló antes, al haberse desarrollado esta herramienta como en un lenguaje de programación convencional (como C), ella es compatible entre plataformas.

VI. RECOMENDACIONES

Al terminar el diseño y desarrollo de este conjunto de herramientas, se plantea una serie de nuevos retos tomando en cuenta el conocimiento adquirido durante este proyecto. Dichos retos permitirán continuar con un proceso que tiene como fin último, el convertir las tecnologías de visualización científica, en herramientas de uso generalizado entre estudiantes, investigadores y público en general.

1. Ampliar el diseño

El conjunto de capacidades que han sido presentadas en el diseño elaborado en este trabajo, representa tan solo un conjunto mínimo de aquellas con las podría contar un sistema de visualización de datos científicos remoto. Muchas nuevas capacidades pueden ser agregadas brindando así una funcionalidad más amplia a dichas herramientas.

Por ejemplo, este diseño ha dejado por fuera cierto tipo de capacidades que tendría un sistema de visualización científico moderno. Entre estas capacidades se encuentra el manejo de diferentes sistemas de coordenadas, las cuales posibilitan un conjunto más amplio de representaciones visuales. También, el uso de técnicas de distorsión no ha sido contemplado en este diseño, aún cuando estas técnicas han demostrado ser de gran ayuda al explorar grandes conjuntos de datos relacionados.

2. Mejorar el conjunto de herramientas

El desarrollo que se realizó no contempló, en su totalidad, las características posibles en un sistema de ejecución remota. Esto se debió en parte, como ya se mencionó, a que los mecanismos que utiliza el OpenDX no facilitan su realización.

Por ejemplo, actualmente el conjunto de herramientas desarrolladas no permite especificar sistemas de coordenadas diferentes a las rectangulares. El uso de coordenadas circulares permitiría ampliar las capacidades de representación de los programas.

3. Desarrollar bajo otros ambientes

Un desarrollo futuro, que traería grandes beneficios, es el de recodificar este diseño en otro tipo de ambientes. Dichos ambientes podrían ser otro tipo de sistemas de visualización científica, como por ejemplo VisAD y VTK, o bien, se podría desarrollar directamente mediante lenguajes de programación tradicional y utilizando las capacidades gráficas que provean dichos lenguajes.

Las ventajas que podrían tenerse, al desarrollar en otros paquetes de visualización científica, es que generalmente dichos ambientes proveen muchas características similares a las de OpenDX. Este no es el caso cuando se trabaja con lenguajes de programación tradicionales con capacidades de gráficos sencillos.

Sin embargo, los lenguajes de programación tradicionales cuentan con la ventaja de una gran flexibilidad a la hora de desarrollar nuevas técnicas de interacción. Esto se debe a que el programador tiene control total sobre el tipo de representación que presenta al usuario y de cómo ésta responderá a las diferentes acciones o eventos que el usuario ejecute sobre dicha representación. En este sentido, el ambiente de programación Java junto con sus librerías de manejo de gráficos, como son el Java2D y el Java3D, parece ser una de las mejores opciones para hacer la recodificación de este diseño.

VI. REFERENCIAS

- Brodlie K., Carpenter L., Earnshaw R.A. (Brodlie y otros, 1991), Scientific Visualization: Techniques and Applications, Springer-Verlag, Alemania, 1991.
- (Carpendale, Cowperthwaite, y Fracchia, 1997) Carpendale, M.S.T., Cowperthwaite, D.J., y Fracchia, F.D. Extending Distortion Viewing from 2D to 3D. IEEE Computer Graphics and Applications, Julio/Agosto, 1997.
- (Carr, Jog y Kumar, 1994) Carr, D.A., Jog, N., y Kumar, H.P. Using Interaction Objects Graphs to Specify and Develop Graphical Widgets, Technical Report, University of Maryland, 1994.
- (Cleveland, 1993) Cleveland, W.S. Visualizing Data, Hobart Press, 1993.
- (Cleveland y McGill, 1988) Cleveland, W.S. y McGill, R. Dynamic Graphics for Statistics, Pacific Grove, CA: Wadsworth and Brooks/Cole, 1988.
- (DeFanti, Brown, y McCormick, 1989) DeFanti, T.A., Brown, M.D., y McCormick, B.H. Visualization: Expanding Scientific and Engineering Research Opportunities, IEEE Computer, Vol.22, No. 8. 1989.
- (Earnshaw y Wiseman, 1992) Earnshaw, R.A. y Wiseman, N. An Introductory Guide to Scientific Visualization, Springer-Verlag, Alemania, 1992
- (Eick, Steffen, y Sumner, 1992) Eick, S.G., Steffen, J.L., y Sumner, E.E., Seesoft: A Tool for Visualizing Line Oriented Software Statistics, IEEE Transactions on Software Engineering, Noviembre, 1992.
- (Feiner y Beshers, 1990) Feiner, S.K. y Beshers, C. Visualizing n-Dimensional Virtual Worlds with n-Vision. Computer Graphics, 24(2) 1990.
- (Goodchild, 1992) Goodchild, M.F. Geographical data modeling. Computers and Geosciences, Vol 4. No.18, 1992
- (Haber y McNabb, 1990) Haber, R. B. y McNabb, L. J. Visualization in Engineering Mechanics: Techniques, Systems and Issues, SIGGRAPH'88, 1988

- (McCormick y DeFanti, 1987) McCormick, B.H. y DeFanti, T.A. Visualization in Scientific Computing, Computer Graphics, Noviembre, 1987
- (McDonald, 1990) McDonald, J.A. Painting Multiple Views of Complex Objects. Proceedings of ECOOP/OOPLSLA'90, 1990.
- (O'Rourke, 1994) O'Rourke, J. Computational Geometry in C. New York: Cambridge University Press, 1994.
- (Preparata y Shamos, 1985) Preparata, F.P. y Shamos, M.I. Computational Geometry. Springer-Verlag, New York, 1985.
- (Robertson, Card y Mackinlay, 1989) Robertson, G.G., Card, S.K. y Mackinlay, J.D. The Cognitive Co-processor for Interactive User Interfaces. Proceedings of UIST'89, ACM Symposium on User Interface Software and Technology. 1989.
- (Roth y Mattis, 1990) Roth, S.F. y Mattis, J. Data Characterization for Intelligent Graphics Presentation, Proceedings of CHI'90, ACM Conference on Human Factors in Computing Systems, New York, 1990.
- (Shneiderman, 1994) Shneiderman, B. Dynamic Queries for Visual Information Seeking. IEEE Software, Vol. 11, No. 6., 1994.
- (Spence y Apperley, 1982) Spence, R. y Apperley, M.D. Data Base Navigation: An Office Environment for the Professional. Behavior and Information Technology, Vol. 1, No. 1, 1982.
- (Tweedie, 1997) Tweedie, L. A. Characterizing Interactive Externalizations. Proceedings of CHI'97, ACM Conference on Human factors in Computing Systems, Atlanta, 1997.
- (Tukey, 1977) Tukey, J.W. Exploratory Data Analysis. Addison Wesley, USA, 1977.